



Recherchebericht

Inhaltsverzeichnis

1. Begriffe	2
1.1 Framework.....	2
1.2 XML.....	2
1.3 URI.....	2
1.4 RDF.....	2
1.5 Ontologie.....	3
1.6 OWL.....	3
1.7 SPARQL.....	3
1.8 DBpedia.....	3
1.9 Pattern.....	3
1.10 Corpus.....	4
1.11 Plug-In.....	4
2. Konzepte	4
2.1 Semantic Web.....	4
2.2 Linked Data.....	4
2.3 Java.....	5
2.3 MVC-Architektur.....	5
3. Aspekte	6
3.1 BOA.....	6
3.1.1 Das Prinzip von BOA.....	6
3.1.1 BOA-Pattern.....	7
3.2 Extraktionsmodul FOX.....	7
3.3 Wichtige Frameworks zur Entwicklung von BOA.....	7
3.3.1 Stanford CoreNLP.....	7
3.3.2 Jena.....	8
3.3.3 Lucene.....	8
3.3.4 Log4j.....	8
3.3.5 Spring.....	8
3.3.6 Vaadin.....	8
4. Quellen	9



1 Begriffe

1.1 Framework

Ein Framework ist eine Grundstruktur bzw. ein Programmiergerüst, das dem Programmierer einen Entwicklungsrahmen für die Erstellung seines Anwendungsprogramms zur Verfügung stellt. Insbesondere werden Frameworks in der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen verwendet. Ziel eines Frameworks ist die Wiederverwendbarkeit von "architektonischen Mustern", wodurch Frameworks meist domänenspezifisch oder auf einen bestimmten Anwendungstyp beschränkt sind.

1.2 XML

XML (*eXtensible Markup Language*) ist ein einfaches, universales Format zum Austausch strukturierter Informationen. Während HTML zur Darstellung von Daten geschaffen wurde, steht bei XML der Datenaustausch im Mittelpunkt. Sie dient vor allem dem plattform- und implementierungsunabhängigen Austausch unterschiedlich stark strukturierter Daten zwischen Computern. Mit Hilfe von XML kann man die logische Bedeutung von Daten, Informationen sowie Texten definieren, ähnlich wie die Tabellen- und Spalten-Bezeichnungen in Datenbanken und Tabellenkalkulationen. Außerdem ermöglicht XML im Gegensatz zu HTML die Definition eigener oder zusätzlicher Befehle wie beispielsweise bei der Definition von Makros in der Textverarbeitung.

1.3 URI

Unter *Uniform Resource Identifier* (URI) versteht man eine weltweit eindeutige Referenz auf abstrakte und physische Ressourcen wie beispielsweise Text, Grafik oder Audio. Sie bilden den Oberbegriff für URL (*Uniform Resource Locator*) und URN (*Uniform Resource Name*). In der Regel sind URIs Zeichenfolgen, die aus folgenden fünf Teilen aufgebaut sein können: `scheme` (Schema), `authority` (Anbieter), `path` (Pfad), `query` (Abfrage) und `fragment` (Teil), wobei in jeder URI nur `scheme` sowie `path` vorhanden sein müssen. Mit Hilfe von URIs können Pfade zu einem Dokument, zu einer Email-Adresse oder zu einer anderen Linkadresse, wie beispielsweise „<http://www.yahoo.com>“, bezeichnet werden.

1.4 RDF

Das *Resource Description Framework* (RDF) ist ein vom *World Wide Web Consortium* (W3C) entwickelter Standard zur formalen Beschreibung von Informationen und bildet die Kernkomponente für das Semantic Web. RDF Dokumente werden in Form von Aussagen gebildet, die nach dem Muster „Subjekt – Prädikat – Objekt“ (Tripel) modelliert werden. Dabei bildet die Menge der Tripel einen gerichteten Graphen, den so genannten RDF-Graph. Dadurch sind die Informationen von Maschinen auswertbar und können weiterverarbeitet werden. Grundlage für RDFs bilden bereits vorhandene XML- und URI-Technologien. Doch anders als bei XML bietet RDF die Möglichkeit beliebige materielle sowie immaterielle Gegenstände und Entitäten zu beschreiben (unstrukturierte Daten¹).

¹Daten sind in drei Typen kategorisierbar: 1. strukturierte Daten – weisen gleiche Struktur auf z.B. Datenbank, Dateien; 2. Semi-strukturierte Daten – keine allgemeine Struktur, sondern nur Teil von Strukturinformationen z.B. XML; 3. Unstrukturierte Daten – keine formalisierte Struktur z.B. Dokumente, Texte, Grafiken, Audio



1.5 Ontologie

Eine Ontologie strukturiert Begriffe eines Wissensgebietes durch eine festgelegte Terminologie und Beziehungen zwischen diesen Begriffen. Auch Regeln zum Ableiten zwischen den Begriffen können enthalten sein. Die Ausgangselemente sind meist Klassen, Relationen, Funktionen und Axiome in Form einer Taxonomie². Daraus ist in der Ontologie ein Netzwerk der Begriffe entwickelt, welche Integritätsbedingungen beinhaltet, um die Korrektheit sowie Gültigkeit zu sichern.

1.6 OWL

Mit Hilfe der *Web Ontology Language* (OWL) werden Ontologien anhand einer formalen Semantik erstellt, publiziert und verteilt. Das Prinzip von OWL besteht darin, Anwendungen zu ermöglichen, den Inhalt von Informationen zu verarbeiten, anstatt sie dem Anwender nur zu präsentieren. Durch zusätzliches Vokabular in Verbindung mit formaler Semantik wird OWL, im Vergleich zu XML oder RDF, eine stärkere Interpretationsmöglichkeit von Webinhalten ermöglicht. Damit bildet OWL eine bedeutende Rolle für das Semantic Web.

1.7 SPARQL

Unter SPARQL (*SPARQL Protocol And RDF Query Language*) versteht man eine Graph-basierte Anfragesprache für RDFs, mit dem sich Daten aus RDF-Dokumenten extrahieren lassen. SPARQL baut in Prinzip auf der RDF-Syntax auf und ist aus folgenden drei Teilen aufgebaut: Aus der „*SPARQL Query Language for RDF*“, also die Abfragesprache selbst, aus der „*SPARQL Query Results XML Format*“, in dem das Format der Abfrageergebnisse in XML beschrieben wird, sowie dem in „*SPARQL Protocol for RDF*“ beschriebenen Protokoll für den Austausch von Abfragen und Ergebnissen. Die Abfragesprache an sich kann im Vergleich zu RDF Subjekt, Prädikat und Objekt durch Variablen ersetzen, wodurch ein Graph-Muster entsteht, mit dem sich ein oder mehrere Teilgraphen mit bestimmten Bedingungen finden lassen.

1.8 DBpedia

DBpedia ist ein Gemeinschaftsprojekt der Universität Leipzig, der Freien Universität Berlin und Open Link Software, um strukturierte Informationen aus Wikipedia zu extrahieren und Web-Anwendungen zugänglich zu machen [1]. Es folgt damit der Philosophie des Semantic Web, Daten in für den Computer verständlicher Form bereitzustellen, um später eine maschinelle Datenextraktion und -analyse dieser strukturierten Informationen zu ermöglichen.

1.9 Pattern

Ein Pattern ist eine Lösungsschablone, die für wiederkehrende Entwurfsprobleme in der Softwarearchitektur und Softwareentwicklung genutzt wird. Sie hilft beispielsweise bei der Strukturierung der Klassen während der Entwurfsphase. Ein gutes Pattern sollte ein oder mehrere Probleme zuverlässig lösen, ein erprobtes Konzept bieten, unabhängig von einer konkreten Programmiersprache und Beziehungen aufzeigen sowie die tiefgehenden Strukturen und Mechanismen eines Systems umfassen.

² Sammlung von strukturierten Begriffen eines Vokabulars oder Glossars



1.10 Corpus

Bei einem Corpus bzw. Textkörper handelt es sich um eine große Sammlung von schriftlichen Texten. Ein Corpus wird heute typischerweise in digitaler Form gehalten. Oft kommt es vor, dass Corpora (Plural von Corpus) nach bestimmten wissenschaftlichen Kriterien zusammengestellt sind und dass sie eine bestimmte Art und Anzahl von Texten umfassen. Außerdem kann ein Corpus Texte einer einzigen Sprache (*monolingual corpus*) oder Textdaten verschiedener Sprachen (*multilingual corpus*) enthalten.

1.11 Plug-In

Ein Plug-In stellt ein Erweiterungsmodul bzw. Softwaremodul dar, um die Funktionalität einer Softwareanwendung zu erweitern bzw. zu spezifizieren. In der Regel kann ein Plug-In nicht unabhängig betrieben werden, da es nur ein Stück Software ist. Doch in seltenen Fällen gehören Plugin-Schnittstellen zur Standardausstattung. Allgemein existieren verschiedene Arten von Plugins wie z.B. Audio-, Grafik-, Browser- oder Server-Plugins.

2 Konzepte

2.1 Semantic Web

„The Semantic Web is not a separate Web but an extension to the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.“³

Tim Berners Lee

Das Problem des heutigen World Wide Webs ist die unüberschaubare Menge an unstrukturierten Daten, die im gesamten Web verteilt sind. Es besteht zwar die Möglichkeit diese Daten mit Hilfe des Computers zu durchsuchen, jedoch um nützliche Informationen daraus zu filtern, müssen sie von Menschen gelesen und interpretiert werden. Computer sind nicht bzw. noch nicht in der Lage die Informationen zu verstehen. Sie können die Daten lediglich dem Menschen bereitstellen. Dieses Problem soll daher mit Hilfe des neuen Konzepts Semantic Web gelöst werden. Wie Tim Berners Lee, dem Gründer des WWWs, einst gesagt hatte, soll das Semantic Web nicht als neues Web gesehen werden, sondern eher als Erweiterung des bestehenden WWW. Das allgemeine Ziel ist die semantische Annotation von Datenbeständen im Internet. Dabei sind die Daten des Semantic Webs in strukturierter Form aufbereitet, welche für Maschinen interpretierbar sind. Dies ermöglicht beispielweise eine verbesserte gezielte Suche nach gewissen Informationen durch Suchmaschinen wie Google oder Yahoo. Grundlagen für das Semantic Web bilden so genannte Semantische Technologien wie RDF, OWL oder SPARQL.

2.2 Linked Data

Linked Data beschreibt das Konzept, strukturierte Daten, die über URIs bzw. Links mit anderen strukturierten Daten in Beziehung gesetzt werden, im Internet öffentlich bereitzustellen. Dabei werden alle Dinge oder Personen, die Gegenstand der vorhandenen Informationen sind, mit eindeutigen URLs identifiziert, welche eine sinnvolle Verknüpfung der Daten bzw. das Auslesen der Relationen durch Rückverfolgung der entsprechenden Links erst ermöglichen. Das heißt, es stehen nicht nur mit Werten

³ Zitat von Tim Berners Lee, dem Begründer des World Wide Webs, in der Ausgabe: Tim Berners-Lee, James Hendler and Ora Lassila. *The Semantic Web*. In: Scientific American, Mai 2001.



versehene Attribute der Objekte als Informationen zur Verfügung, sondern auch deren Beziehungen untereinander. Im Unterschied zu den meisten herkömmlichen Informationsnetzwerken können und sollen diese Daten dabei jedoch nicht nur von menschlichen Nutzern, sondern auch von Computerprogrammen eingelesen und verstanden werden. Dies wird durch ihre Strukturiertheit und den logischen Aufbau der zugrundeliegenden Verknüpfungen erreicht.

2.3 Java

Java ist eine objektorientierte Programmiersprache. Entwickelt wurde sie von Sun Microsystems und wurde 1995 das erste Mal vorgestellt. Seitdem wird sie fortwährend weiterentwickelt.

Ein Hauptmerkmal der Sprache ist die Einfachheit, da der Sprachumfang im Vergleich zu anderen objektorientierten Sprachen, wie z.B. C++, gering ist und die Eingewöhnung aus anderen objektorientierten Sprachen durch die Anlehnung an C/C++ Syntax erleichtert wird. Weiterhin ist Java parallelisierbar, also multithreadable und dynamisch, wodurch einzelne Module eines Programms leicht austauschbar bzw. erweiterbar sind. Statt Zeigern gibt es nur Referenzen auf Objekte. Da die eigentliche Speicheradresse dabei weder verändert, noch ausgelesen werden kann, ist Zeigerarithmetik in Java nicht möglich. Auch Mehrfachvererbung wird nicht unterstützt. Diese Eigenschaften dienen der Sicherheit und Stabilität der Sprache, was ein weiteres wichtiges Merkmal von Java ist. Neben diesen Eigenschaften hat Java eine weitere Besonderheit, die Plattformunabhängigkeit. Neben dem System, spielt die Hardware theoretisch keine Rolle bei der Ausführung eines Java-Programms. Dies liegt im Aufbau der Java-Technologie begründet. Dazu gehören das Entwicklungswerkzeug (JDK) und die Java Laufzeitumgebung (JRE). Grundlegender Bestandteil der Laufzeitumgebung ist die Virtuelle Maschine, die den vom Compiler übersetzten Bytecode ausführen kann, was somit unabhängig vom eigentlichen System ist. Java ist damit objektorientiert, leistungsfähig, sicher, stabil und dynamisch und parallelisierbar.

2.4 MVC-Architektur

MVC (*Model-View-Controller*) ist ein bewährtes Architekturmodell für die objektorientierte Software-Entwicklung. Basis für dieses Modell bildet das Entwurfsmuster "Beobachter", in dem ein Objekt von einem oder mehreren anderen Objekten "beobachtet" wird, damit diese auf das überwachte Objekt betreffende Änderungen reagieren können.

Ein nach MVC konzipiertes Programm wird in drei Einheiten unterteilt:

- Das Modell (*Model*): In ihm ist die Kernfunktion der Anwendung realisiert. Es bildet also die Basis des Programms und nimmt im oben erwähnten Muster die Rolle des "Beobachteten" ein.
- Die Präsentation (*View*): Diese Einheit dient vor allem der Präsentation der Daten (Model). Auf das Schema bezogen steht sie für den "Beobachter", der vom "Beobachteten" bei Änderung in Kenntnis gesetzt wird und sich aktualisiert. Dabei können mehrere unterschiedliche Präsentationen für ein Programm gleichzeitig ausgeführt werden.
- Die Programmsteuerung (*Controller*): Die dritte Komponente, der Controller, übernimmt die Steuerung der Modellmanipulation und bildet zusammen mit der Präsentation die Nutzerschnittstelle. Sie nimmt Benutzereingaben entgegen und wertet diese, entsprechend im Quellcode festgelegter Regeln, aus. Dabei entscheidet sich, welche Teilmengen der vorhandenen Daten auf welche Art und Weise geändert werden sollen.

Diese Definition beschreibt MVC, wie es in Smalltalk eingeführt wurde. Jedoch gibt es ein weiteres Modell, welches in Webapplikationen seine Anwendung findet. Dort sind die Aufgaben der einzelnen Einheiten entsprechend angepasst:

Der View ist eine JSP (*JavaServer Pages*)-Seite, welche die Daten repräsentiert und zugleich Eingaben (vorher Aufgabe vom Controller) vom Benutzer entgegennimmt. Der Controller ist ein *Servlet*, der von der JSP aufgerufen wird, um die Eingabe zu interpretieren und die Ausführung der entsprechenden Befehle zu steuern. Als Modell wird meist ein *Java Bean* benutzt, welche relevante Daten für das Modell enthält.

3 Aspekte

3.1 BOA

3.1.1 Das Prinzip von BOA

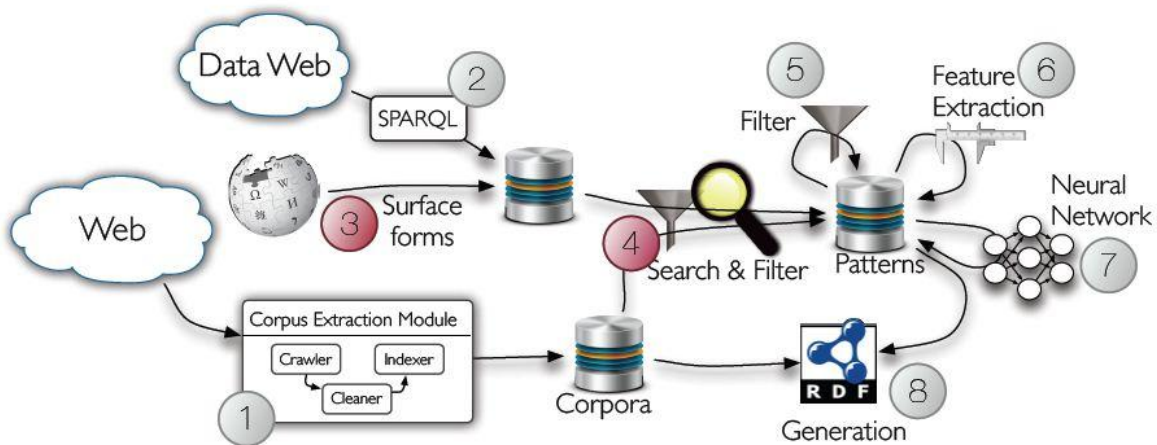


Abb. 3.1.1: Überblick zur Funktionsweise von BOA

BOA (**BO**otstrapping **linked data**) ist ein System, um strukturierte Daten iterativ aus natürlicher Sprache zu extrahieren. Im Gegensatz zu vergleichbaren Projekten greift es auf das Linked Data Web als Wissensbasis zurück und nutzt maschinelles Lernen zum bewerten der hinzugewonnenen Daten. Darüber hinaus bietet es die Möglichkeit Oberflächenformen (also unterschiedliche Arten der Darstellung von Informationen im Text) für die Wissensbasis aus der Wikipedia zu generieren. Die Sprache der Texte, aus denen sie extrahiert werden soll, ist dabei grundsätzlich unerheblich, es können jedoch sprachspezifische Merkmale verwendet werden.

Der Extraktionsprozess von BOA beginnt mit crawlen (aufsammeln von Informationen) der Quelle, sowie dem säubern und indizieren der erhaltenen Textfragmente, woraus ein Corpus gewonnen wird. Mit Hilfe der Wissensbasis wird der Corpus durchsucht und gefiltert. Dabei wird nach Paaren von Begriffen gesucht, die über ein Prädikat aus der Wissensbasis verknüpft sind, und deren gemeinsames Vorkommen statistisch analysiert. Die Bewertung eines solchen Musters übernimmt ein überwachtes neuronales Netzwerk (also eine künstliche Intelligenz), welches mit zuvor manuell gekennzeichneten Mustern trainiert wird. Aus den Mustern mit den besten Bewertungen werden schließlich RDF-Aussagen gewonnen. Diese können anschließend mit der Wissensbasis vereinigt werden, um im nächsten Durchlauf des Prozesses wieder verwendet zu werden.



3.1.2 BOA-Pattern

Die Extraktion von Textsegmenten wird mit Hilfe des BOA-Patterns realisiert. Dabei werden allgemein folgende drei Schritte benötigt⁴:

- (1) Gegebene Menge von Entitäten **s** (Subjekt) und **o** (Objekt), die durch **p** (Prädikat) verbunden werden.
- (2) Finde alle Sätze, welche **s** und **o** enthalten.
- (3) Ersetze alle Labels mit Variablen z.B. ?D? oder ?R?

Schließlich können so genannte *Pattern-Mappings* erstellt werden. Darüber hinaus sollte ein BOA-Pattern auch Eigenschaften wie z.B. Support, Specificity und Typicity erfüllen (siehe in [GerN⁺11]).

3.2 Extraktionsmodul FOX

Da das BOA-Framework die Möglichkeit bietet Extraktionsmodule als Plugins zu integrieren, soll im Folgenden FOX als Beispiel kurz vorgestellt werden.

FOX (*Federated knOwledge eXtraction Framework*) ist ein Projekt, das von der Forschungsgruppe AKSW der Universität Leipzig entwickelt wurde. Es ist ein Framework, das die Linked Data Cloud integriert und die Vielfalt von NLP (*Natural Language Processing*) Algorithmen anwendet, um RDF Tripel mit hoher Exaktheit aus natürlichen Sprachen zu extrahieren. Es kombiniert verschiedene Tools für den Prozess einer natürlichen Sprache und vereinigt deren Ergebnisse durch effiziente gleichartige bzw. gemeinsame Lernalgorithmen. Die Architektur ist dabei in drei Hauptteile gegliedert:

- Der erste Hauptteil ist die „Machine Learning“- Schicht, welche Schnittstellen für angepasste Lerntechniken implementiert. Diese Schicht besteht aus zwei Hauptmodulen - dem Trainingsmodul und dem Prädiktionsmodul.
- Der zweite Hauptteil von FOX ist die „Controlling“-Schicht, welche den Zugriff auf die Module der NLP koordiniert.
- Der dritte und letzte Hauptteil ist die „Tool“-Schicht, in der alle von FOX integrierten NLP-Tools und Anwendungen enthalten sind. Während der Lernphase kann FOX diese Tools nicht anwenden, da sie noch unbekannt sind. Vielmehr versucht FOX, basierend auf den bereits in den Tools enthaltenen Modellen, die geeignetste Kombination von Klassen und Kategorien zu finden.

3.3 Wichtige Frameworks zur Entwicklung von BOA

3.3.1 Stanford CoreNLP

Stanford CoreNLP ist eine Java-Bibliothek zur Analyse natürlicher, englischsprachiger Texte. Die Bibliothek versieht dazu einzelne Text- und Satzbestandteile mit sogenannten Annotationen und kann daraus Graphen erstellen. Dabei kann man auf eine Reihe weitestgehend frei kombinierbarer Module zurückgreifen, die unterschiedliche Annotationen generieren. Die Funktionalität umfasst unter anderem das Säubern von Text, die Analyse des Satzbaus und das Erkennen von Namen und Daten aus einer Vielzahl von Wissensgebieten.

⁴ In Präsentation: *BOA - Extracting Multilingual Natural-Language Patterns for RDF Predicates* von Daniel Gerber
<http://bis.informatik.uni-leipzig.de/DanielGerber>



3.3.2 Jena

Jena ist ein frei zugängliches Java-Framework, welches explizit für die Anwendung in semantischen Netzen entwickelt wurde. Es unterstützt OWL und bietet dem Programmierer eine Schnittstelle, um RDF Graphen auszulesen (SPARQL) oder bearbeiten (SPARUL) zu können. Einen Überblick zur Jena API findet man unter http://jena.sourceforge.net/tutorial/RDF_API/index.html.

3.3.3 Lucene

Lucene ist eine Bibliothek zur Volltextsuche, die in Java geschrieben wurde. Sie stellt Klassen und Funktionen für eigene Suchmaschinen zur Verfügung. Sie ist plattformunabhängig und besonders schnell und effizient. Zum Suchen wird zuerst der Datenbestand, der durchsucht werden soll, indiziert, d.h. die Daten werden unter angegebenen Suchparametern durchsucht und in einem Index gespeichert (ähnlich wie bei einem Schlagwortverzeichnis, bei dem für ein Wort relevante Seiten im Buch aufgelistet sind). Danach wird mit Hilfe der Indizes und einem vorher festgelegten Suchalgorithmus gesucht.

3.3.4 Log4j

Log4j ist ein Logging-Framework, das sich mittlerweile zum gängigen Standard vieler Softwareprojekte entwickelt hat. Fehler und Informationen werden mit log4j nicht einfach nur ausgegeben, sondern über Logger an ein zugewiesenes System weitergeleitet. Dabei werden unwichtige Meldungen unter Umständen nicht weitergeleitet und stören so den Programmablauf nicht. Also ist zunächst nur die Einstufung der Wichtigkeit einer Meldung für die Programmierung von Bedeutung, nicht aber deren Verarbeitung. Zur Laufzeit wird entschieden wie die Ausgabe bzw. Verarbeitung der Fehler geschieht. Appender definieren dabei den Ausgabekanal von Meldungen. Ausgaben sind z.B. in Form von E-Mails, Sockets, Dateien oder Konsolen möglich, wodurch eine hohe Verarbeitungsgeschwindigkeit gesichert wird. Das Loggen selbst verlangsamt also nicht die Systemleistung. Als Alternative ist bereits in der Java-Klassenbibliothek Java-Logging seit 1.4 Bestandteil.

3.3.5 Spring

Spring ist ein eigens für die Java-Plattform entwickeltes, quelloffenes Framework, dessen Ziel die vereinfachte Entwicklung von Java/Java EE Applikationen und die Förderung guter Programmierpraktiken ist. Es steht die Entkopplung der Applikationskomponenten im Vordergrund. Dies wird unter anderem gewährleistet durch das Zusammenspiel unterschiedlichster Plattformen und Tools, von JavaEE-Servern über Persistenztools bis hin zur Web-Integration. Des Weiteren ermöglicht eine mit Spring erstellte Applikation eine bessere Trennung der Module und die Möglichkeit in einem laufenden System Module zu entfernen, hinzuzufügen oder zu aktualisieren.

3.3.6 Vaadin

Vaadin ist ein freies modernes Java Webanwendungs-Framework. Anders als bei JavaScript-Bibliotheken und auf Browser-Plugins basierenden Lösungen bietet Vaadin eine serverseitige Architektur, d.h. ein Großteil der Programmlogik läuft auf einem Server ab. Dabei wird auf Browser-Seite mit *Ajax* gearbeitet und auf Client-Seite kommt das *Google Web Toolkit* (GWT) zum Einsatz, das für die Darstellung der Webseiten genutzt wird. Außerdem gibt es die Möglichkeit die Standardkomponenten von Vaadin mit eigenen GWT-Steuerelementen zu erweitern, wodurch sich Themes mit CSS (*Cascading Style Sheets*) erstellen lassen.



4 Quellen

Literatur

- [Bad06] Guido Badertscher. *Generelle Queryverarbeitung für das Semantic Web – Effiziente Suche in strukturierten Daten*. VDM Verlag Dr. Müller, Saarbrücken, 2006.
- [GerN⁺11] Daniel Gerber, Axel-Cyrille Ngonga Ngomo. *Extracting Multilingual Natural-Language Patterns for RDF Predicates*. In: Web Scale Knowledge Extraction Workshop, ISWC, 2011.
- [HKRS08] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure. *Semantic Web Grundlagen, Erste Auflage*. Springer-Verlag, Berlin Heidelberg, 2008.
- [N⁺HeSHi] Axel-Cyrille Ngonga Ngomo, Norman Heino, René Speck, Stanley Hillner. *Federated Knowledge Extraction for Semantic Web Applications*.
- [Wisch06] Gabriele Wichmann. *Entwurf Semantic Web – Entwicklung, Werkzeuge, Sprachen*. VDM Verlag Dr. Müller, Saarbrücken, 2006.

Internet

- [1] Wikipedia <http://de.wikipedia.org/> für folgende Begriffe/Stichworte:
(Stand: 09.01.2012) Resource Description Framework, Uniform Resource Identifier, Framework, Extensible Markup Language
(Stand: 10.01.2012) Ontologie, Entwurfsmuster (Pattern), SPARQL
(Stand: 11.01.2012) Linked Data
(Stand: 12.01.2012) Web Ontology Language, Textkorpus, Plug-In, DBpedia, MVC
(Stand: 15.01.2012) Log4j, Vaadin
(Stand: 16.01.2012) Jena
- [2] <http://www.itwissen.info/definition/lexikon/Framework-framework.html> (Stand: 10.01.12)
- [3] http://www.gi.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/ontologien-57.html (Stand: 10.01.12)
- [4] <http://www.michaels-website.de/wp-content/uploads/2008/10/sparql.pdf> (Stand: 10.01.2012)
- [5] <http://www.w3.org/TR/owl-features/> (Stand: 12.01.2012)
- [6] <http://www.drweb.de/magazin/was-ist-lexikon-plugin/> (Stand: 13.01.2012)
- [7] <http://lots.uni-leipzig.de/xquery/tutorial/xml/definition/> (Stand: 13.01.2012)
- [8] <http://www.boku.ac.at/htmlinf/xmlkurz.html> (Stand: 13.01.2012)
- [10] http://openbook.galileocomputing.de/oo/oo_06_moduleundarchitektur_001.htm (Stand: 13.01.2012)
- [11] http://www.altova.com/de/semantic_web.html (Stand: 14.01.2012)
- [12] <http://wap.ecw.de/archives/1366> (Stand: 15.01.2012)
- [13] <http://www.springsource.org/osgi> (Stand: 15.01.2012)
- [14] <http://www.fh-wedel.de/~si/seminare/ws02/Ausarbeitung/e.lucene/1.html> (Stand: 15.01.2012)
- [15] <http://www.torsten-horn.de/techdocs/java-log4j.htm> (Stand: 15.01.2012)
- [16] <http://nlp.stanford.edu/software/corenlp.shtml> (Stand: 15.01.2012)
- [17] <http://aksw.org/Projects/FOX> (Stand: 15.01.2012)

Abbildungen

1. Abb.3.1.1: In Präsentation und Paper: *BOA - Extracting Multilingual Natural-Language Patterns for RDF Predicates* von Daniel Gerber <http://bis.informatik.uni-leipzig.de/DanielGerber>